

06-01-00

A

Docket No.: M-8202 US

May 31, 2000

05/31/00
JC816 U.S. PTO

JC515 U.S. PTO
09/583966
05/31/00

Box Patent Application
Assistant Commissioner for Patents
Washington, D. C. 20231

Enclosed herewith for filing is a patent application, as follows:

Inventors: Hall, Robert A.; Kolecki, Stephen P.
Title: Buffer Status In An Asymmetrical Gap Environment
X Return Receipt Postcard
X This Transmittal Letter (in duplicate)
17 page(s) Specification (not including claims)
15 page(s) Claims
1 page Abstract
5 Sheet(s) of Drawings
3 page(s) Declaration For Patent Application and Power of Attorney

CLAIMS AS FILED

For	Number Filed		Number Extra		Rate		Basic Fee
Total Claims	58	-20 =	38	x	\$18.00 =	\$	\$690.00 684.00
Independent Claims	8	-3 =	5	x	\$78.00 =	\$	390.00
<input type="checkbox"/>	Fee of _____ for the first filing of one or more multiple dependent claims per application						\$
<input type="checkbox"/>	Fee for Request for Extension of Time						\$

Please make the following charges to Deposit Account 19-2386:

- ☒ Total fee for filing the patent application in the amount of \$ 1,764.00
☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account 19-2386.

EXPRESS MAIL LABEL
NO:

EL514816150US

Respectfully submitted,

Dale R. Cook
Attorney for Applicants
Reg. No. 42,434

25 Metro Drive, Suite 700
San Jose, CA 95110
Phone 408 453-9200
Fax 408 453-7979

Austin, TX
Newport Beach, CA
San Francisco, CA

"Express Mail" mailing label number:

EL514816150US

BUFFER STATUS IN AN ASYMMETRICAL GAP ENVIRONMENT

Robert A. Hall
Stephen P. Kolecki

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates, in general, to data communications.

Description of the Related Art

A data communications network is the interconnection of two or more communicating entities (i.e., data sources and/or sinks) over one or more data links. A data communications network allows communication between multiple communicating entities over one or more data communications links.

One type of data communications network is the Synchronous Optical Network (SONET). Generally (although not always), SONETs are composed of network nodes strung together in a ring format.

A SONET has several relatively unique features which distinguishes it from other types of data communications networks. One unique feature is that an attempt is made to synchronize all data nodes in a SONET to the same clock system. However, insofar as SONET nodes are typically dispersed over a large geographic area, the clock signals received by each SONET node generally vary from each other in phase (e.g., due to different propagation delays arising from varying distances from a central clock) and/or frequency (e.g., when an attempt is made to synchronize various parts of a SONET to different clocks making up a clock system, where each clock in the system is supposedly synchronized in phase and frequency with every other clock but which in reality vary slightly due to physical variations). Consequently, it is not uncommon for the signals on the SONET node ingress and egress data links to vary slightly from each other either in phase and/or frequency. Consequently, rather than referring to such signals as truly synchronous, it is common in the art to refer to such signals as plesiochronous (i.e., generated by SONET nodes with separate network clocks which vary only slightly in phase and or frequency).

Since the signals on the ingress and egress data links of a SONET node can vary in phase and/or frequency, SONET does not merely rely on timing to ensure that data is correctly transported across nodes. Rather, SONET uses a data structures (called pointers in the SONET nomenclature) to compensate for frequency and phase variations across nodes. (In SONET, a pointer is simply an offset value that points to the byte where a SPE (synchronous payload envelope, which can be thought of as "user data") begins within a data structure known as a standard SONET frame. If relative severe timing differences exist between the ingress and egress data links of a SONET node, "positive stuffing" (e.g., inserting a byte) or negative stuffing (e.g., removing a byte) can be used at each SONET node to adjust for the phase and/or frequency delays, and the pointer adjusted accordingly to ensure that the next node receiving the SONET data structure can discern where a SPE resides.

In practice, since it is not known in advance whether or not the phase and/or frequency difference between ingress and egress data links of a SONET node is severe enough to warrant stuffing (positive or negative), it is common within the art to buffer received SPE data, and engage in positive or negative stuffing dependent upon whether the buffer is almost full or almost empty. If positive stuffing is used the pointer is incremented and if negative stuffing is used the pointer is decremented, thereby allowing correct discernment of the SPE at the next node. The foregoing is explained in brief overview in relation to Figure 1.

With reference now to Figure 1, shown is a partial block diagram of typical SONET node 100. Depicted are pointer interpreter 102, FIFO (first in, first out - the first data written into the buffer will be the first data read out of the buffer, which is illustrated via FIFO buffer data-out line 150 being connected directly to the FIFO buffer slot 1 and FIFO buffer data-in line 152 being connected to the first FIFO buffer slot which is empty - data-in line 152 is meant to indicate that SPE data is written to the first empty slot of the buffer; if data is read from the buffer the first empty slot of the buffer will appear closer to FIFO buffer slot 1 (e.g., data-in line 152 would appear in the figure to move up one slot toward buffer slot 1) and as data is written to the buffer the first empty slot of the buffer would appear closer to the last FIFO buffer slot 16 (e.g., data-in line 152 would appear in the figure to move down one slot

toward buffer slot 1)) buffer 104, and pointer generator 106. Illustrated are high-level diagrams of standard SONET frames 108 and 110. Shown is that SONET frame 108 enters standard SONET node 100 via ingress data link 112. Depicted is that SONET frame 110 leaves standard SONET node 100 via egress data link 114.

SONET frames 108 and 110 show that each row of each SONET frame has 3 columns of overhead (data utilized to ensure that the SONET works correctly, and which is generally referred to in the art as a 3 column "transport gap," since it represents a gap in the data being transported) and 87 columns of payload (data transmitted through the SONET by SONET users). Those skilled in the art will recognize that the size of each "column" will typically vary dependent upon the number of STS (synchronous transport signals) in use. For example, when one STS is in use (denoted in the art via the symbology "STS-1") each column is 1 byte wide; when two STSes are in use (denoted in the art via the symbology "STS-2"), each column is two bytes wide; when 3 STSes are in use (denoted in the art via the symbology "STS-3"), each column is three bytes wide; etc. In general, the number of bytes per column is a function of the number of STSes in use "N," where N is some positive integer; for example, for STS-48 (i.e., N = 48) each column of each SONET frame would be 48 bytes wide.

For ease of understanding, the discussion from this point forward will generally describe embodiments via use of exemplary STS-1 SONET frames. However, those skilled in the art will appreciate that the discussion herein generalizes to virtually any STS-N SONET frame via multiplying the size of the data structures described herein by the number, N, of STSes in use in any particular implementation.

Referring now again to Figure 1, notice that FIFO buffer 104 is shown as being finite in size (e.g., as having 16 slots). Hence, pointer interpreter 102 needs to be apprised as to when FIFO 104 is "almost full" so that pointer interpreter 102 does not inadvertently overwrite data in FIFO 104. If pointer interpreter 102 and pointer generator 106 find that FIFO 104 is almost full, pointer interpreter 102 and pointer generator 106 engage in appropriate action to ensure that data is not overwritten in FIFO 104 and that pointer generator 106 adjusts the pointer value sufficient to reflect the fact that at least one byte will not be written into FIFO 104 (e.g., the pointer is decremented to reflect negative stuffing).

If pointer interpreter 102 and pointer generator 106 find that FIFO 104 is almost empty, pointer interpreter 102 and pointer generator 106 engage in appropriate action to ensure that data is written into FIFO 104 sufficient such that data always exists in FIFO 104 sufficient such that a constant outflow of data can be maintained on egress line 114 and such that that pointer generator 106 adjusts the pointer value sufficient to reflect the fact that at least one extra byte will be written into FIFO 104 (e.g., the pointer is incremented to reflect the positive stuffing).

It has been discovered by the inventors named herein (the inventors), and such discovery forms part of the inventive content herein, that - irrespective of the fact that pointer interpreter 102 and pointer generator 106 view FIFO buffer 104 from the different standpoints of FIFO buffer data-in line 152 and FIFO buffer data-out line 150, respectively -- pointer interpreter 102 and pointer generator 106 can achieve appropriate positive or negative stuffing using the same indicators of "almost empty" or "almost full" full buffers since both pointer interpreter 102 and pointer generator 106 are keyed to standard SONET frame formats. It has been discovered by the inventors, and such discovery forms part of the inventive content herein, that if either pointer interpreter 102 or pointer generator 106 is keyed to a non-standard SONET frame format (e.g., such as is described in copending application, attorney docket number M-7166-1P US (previously M-7268 US)entitled Method And Apparatus For A Rearrangeably Non-Blocking Switching Matrix, naming Saleh, Ali Najib; Duschatko, Douglas E.; and Quibodeaux, Lane Byron as inventors, having serial number 09/477,166, filed 1/4/00 and hereby incorporated by reference in its entirety), severe difficulties arise with respect to determining the almost empty and almost full buffer conditions, which consequently give rise to difficulties in performing appropriate negative and/or positive stuffing.

Summary

The inventors named herein have invented a method and system which allow for appropriate positive and negative stuffing when either or both a pointer interpreter or a pointer generator are keyed to a non-standard SONET frame format.

In one embodiment, a method includes but is not limited to keying a buffer status to a transport gap other than a standard SONET transport gap. In various embodiments, hardware and/or software are utilized to effect the foregoing-described method. In one embodiment, the foregoing referenced buffer status is that of a transmit buffer. In one embodiment, the foregoing referenced buffer status is that of a receive buffer.

In one embodiment, a method includes but is not limited to detecting a transition involving at least one SONET frame; engaging in negative stuffing, in response to the detecting yielding a determination that a receive FIFO buffer is almost full during the transition involving at least one SONET frame; and engaging in positive stuffing, in response to the detecting yielding a determination that a receive FIFO buffer is almost empty during the transition involving at least one SONET frame. In various embodiments, hardware and/or software are utilized to effect the foregoing-described method.

In one embodiment, a method includes but is not limited to detecting a transition involving at least one SONET frame; engaging in negative stuffing, in response to said detecting yielding a determination that a transmit FIFO buffer is almost full during the transition involving at least one SONET frame; and engaging in positive stuffing, in response to said detecting yielding a determination that a transmit FIFO buffer is almost empty during the transition involving at least one SONET frame. In various embodiments, hardware and/or software are utilized to effect the foregoing-described method.

In one embodiment, a SONET node includes but is not limited to at least one pointer interpreter having an almost full buffer detector set substantially equal to a number of columns present in a non-standard SONET transport gap. In one embodiment, a SONET node includes but is not limited to at least one pointer generator having an almost empty buffer detector set substantially equal to a number of columns present in a non-standard SONET transport gap. In one embodiment, a SONET node includes but is not limited to one or more components selected from the group comprising a processor, a memory device, a bus, and a communications device. In one embodiment, a SONET node includes but is not limited to a Cisco Systems ONS (Optical Network System) 15900 Series Wavelength Router.

The foregoing is a summary and thus contains, by necessity, simplifications, generalizations and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of the present invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Figure 1 shows a partial block diagram of typical SONET node 100.

Figure 2 depicts the partial block diagram of typical SONET node 100 of Figure 1, modified to become modified SONET node 200.

Figure 3 illustrates an expanded partial view of the "receive" side of switch logic 202 wherein it is shown that in one embodiment pointer interpreter 208 treats as "almost full" FIFO buffer 210 when it is twenty-seven (27) or less slots (where each slot is of size sufficient to hold at least one column of SPE data) away from being completely full.

Figure 4 shows an expanded partial view of the "transmit" side of switch logic 202 wherein it is shown that in one embodiment pointer interpreter 214 treats as "almost full" FIFO buffer 210 when it is three (3) or less slots (where each slot is of size sufficient to hold at least one column of SPE data) away from being completely empty; also shown is that in one embodiment "almost full" is set as five (5) slots rather than three (3), for reasons of practicality.

Figure 5 illustrates a pictorial representation of a conventional data processing system which can be utilized in accordance with illustrative embodiments of embodiments described herein.

The use of the same reference symbols in different drawings indicates similar or identical items.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

The following sets forth a detailed description of the best contemplated mode for carrying out the independent invention(s) described herein. The description is intended to be illustrative and should not be taken to be limiting.

Referring now to Figure 2, shown is the partial block diagram of typical SONET node 100 of Figure 1, modified to become modified SONET node 200. Depicted is that, with respect to other SONET nodes interacting with modified SONET node 200 over ingress data link 112 and egress data link 114, modified SONET node 200 appears to function in substantially the same fashion as unmodified SONET node 100.

Illustrated is that internal to modified SONET node 200, the functioning is radically different from that of unmodified SONET node 100. Shown is that switch logic 202 operates on and functions with non-standard SONET frames 204 and 206. Shown is that modified SONET node 200 uses non-standard SONET frames -- such as non-standard SONET frames 204 and 206 -- internal to itself, and yet appears to SONET nodes communicating over ingress data link 112 and egress data link 114 "as if" it were an unmodified SONET node. That is, modified SONET node 200 still receives standard SONET frames, such as standard SONET frame 108, over ingress data link 112 and still transmits standard SONET frames, such as standard SONET frame 110, over egress data link 114.

Making modified SONET node 200 appear "as if" it were unmodified SONET node 100 is achieved on the "receive" side of switch logic 202 (associated with ingress data link 112) by constructing non-standard SONET frames, such as non-standard SONET frame 204, from received standard SONET frames, such as standard SONET frame 108. Making modified SONET node 200 appear "as if" it were unmodified SONET node 100 is achieved on the "transmit" side of switch logic 202 (associated with ingress data link 112) by constructing standard SONET frames, such as standard SONET frame 110, from non-standard SONET frames, such as non-standard SONET frame 206.

On the "receive" side of switch logic 202 this is accomplished via pointer interpreter 208 and pointer generator 212. Pointer interpreter 208 interprets the overhead columns of standard SONET frame 108 and writes the payload (SPE) columns of standard SONET frame 108 into FIFO buffer 210. Pointer generator 212 interacts with pointer interpreter 208 to obtain the

27 columns of overhead data (3 columns/row x 9 rows/SONET frame x 1 byte/column (for STS-1 frames) yields 27 columns of overhead data/SONET frame - for STS-1; numbers would be multiplied by factor of N other STSes, such as STS-N were in use) in order to construct the 27 column overhead data structure of non-standard SONET frame 204 (which, since it also is representative of a gap in the payload data, can also be viewed as a "transport gap," in a fashion analogous to the way the 3 column structure "transport gap" of the related art is viewed), to which is appended the payload data of standard STS frame 108.

It has been discovered by the inventors that since pointer interpreter 208 and pointer generator 212 are keyed to different frame structures (e.g., standard SONET frame 108 and non-standard SONET frame 204 respectively), what constitutes an "almost full" or "almost empty" FIFO buffer 210 will vary dependent upon whether FIFO buffer 210 is being viewed from the standpoint of pointer interpreter 208 or pointer generator 212. For example, since pointer interpreter 208 tends to cooperate in writing 87 column "chunks" of payload data (e.g., the number of columns between overhead columns) to FIFO buffer 210, what constitutes "almost empty" and "almost full" from the standpoint of pointer interpreter 208 will be different from that seen by pointer generator 202, which tends to read out the entire payload contents from FIFO buffer 210 en mass subsequent to the construction of the 27 byte overhead data structure of non-standard SONET frame 204.

That is, insofar as pointer generator 212 generally processes all 27 columns of overhead data prior to reading any payload data from FIFO buffer 210, in order to be sure that data in FIFO buffer 210 is not overwritten, pointer interpreter 208 should treat the "almost full" condition as that condition in which FIFO buffer 210 is at least 27 bytes away from being full, in order to account for the eventuality that pointer interpreter 208 will need to write 27 columns of SPE data while pointer generator 212 is in the process of constructing the 27 column overhead data structure of non-standard SONET frame 204 (and hence not reading data from FIFO buffer 210), so that pointer interpreter 208 can prepare to do negative stuffing should the need arise.

Conversely, insofar as pointer generator 212 is concerned, there could be instances in which pointer interpreter 208 is reading three columns of overhead data and not writing data to FIFO buffer 210. Consequently, in order to be safe, pointer generator 212 preferably should treat the "almost empty" condition as being 3 or less columns of data away

from a completely empty FIFO buffer 210 (which for an STS-1 frame equates to 3 bytes); however, in at least one implementation it has been empirically determined that due to practical considerations it is useful to have the "almost empty" indicator set to 5 columns (which for an STS-1 frame equates to 5 bytes).

Referring now to Figure 3 shown is an expanded partial view of the "receive" side of switch logic 202 wherein it is shown that in one embodiment pointer interpreter 208 treats as "almost full" FIFO buffer 210 when it is twenty-seven (27) or less slots (where each slot is of size sufficient to hold at least one column of SPE data) away from being completely full; also shown is that in one embodiment almost full is treated as twenty-eight (28) slots or less for reasons of practicality. Also shown is that in one embodiment pointer generator 212 treats as "almost empty" FIFO buffer 210 when it is three (3) or less slots away from being completely empty; also shown is that in one embodiment "almost empty" is set as five (5) slots rather than three (3), for reasons of practicality.

Referring again to Figure 2, on the "transmit" side of switch logic 202 modified SONET node 200 is made to appear "as if" it were unmodified SONET node 100 via the activities of pointer interpreter 214 and pointer generator 218. Pointer interpreter 214 interprets the overhead columns of non-standard SONET frame 206 and writes the payload (SPE) columns of non-standard SONET frame 206 into FIFO buffer 216. Pointer generator 218 interacts with pointer interpreter 214 to obtain the 27 bytes of overhead data (3 columns/row x 9 rows/SONET frame x 1 byte/column (for STS-1 frames) yields 27 columns of overhead data/SONET frame - for STS-1; numbers would be multiplied by factor of N other STSes, such as STS-N were in use) in order to construct the 3 columns x 9 row overhead data structures of standard SONET frame 110, where each 3 column overhead data structure has appended to it 87 columns of payload data for the row to which each 3 column overhead data structure belongs.

It has been discovered by the inventors that since pointer interpreter 214 and pointer generator 218 are keyed to different frame structures (e.g., non-standard SONET frame 206 and standard SONET frame 110 respectively), what constitutes an "almost full" or "almost empty" FIFO buffer 216 will vary dependent upon whether FIFO buffer 216 is being viewed from the standpoint of pointer interpreter 214 or pointer generator 218. For example, since pointer generator 218 tends to cooperate in reading 87 column "chunks" of payload data (e.g., the number of columns between

overhead columns), what constitutes "almost empty" and "almost full" from the standpoint of pointer generator 218 will be different from that seen by pointer interpreter 214, which tends to interpret the 27 byte overhead data structure of non-standard SONET frame 206 prior to writing any data into FIFO buffer 216.

That is, insofar as pointer interpreter 214 generally processes all 27 columns of overhead data prior to writing any payload data to FIFO buffer 216, in order to ensure that loss of data continuity does not occur, pointer generator 218 should treat the "almost empty" condition as that condition which FIFO buffer 216 is at least 27 bytes away from being empty, in order to account for the eventuality that pointer generator might be interpreting the 27 byte overhead data structure of non-standard SONET frame 206 at some particular interval in time during which pointer generator 218 is reading data, so that pointer generator 218 can prepare to do positive stuffing should the need arise.

Conversely, insofar as pointer interpreter 214 is concerned, there could be instances in which pointer interpreter 208 is constructing three columns of overhead data and not reading data from FIFO buffer 216. Consequently, in order to be safe, pointer interpreter 214 preferably should treat the "almost full" condition as being 3 or less columns of data away from a completely empty FIFO buffer 216 (which for an STS-1 frame equates to 3 bytes); however, in at least one implementation it has been empirically determined that due to practical considerations it is useful to have the "almost empty" indicator set to 5 columns, which for an STS-1 frame equates to 5 bytes.

Referring now to Figure 4, shown is an expanded partial view of the "transmit" side of switch logic 202 wherein it is shown that in one embodiment pointer interpreter 214 treats as "almost full" FIFO buffer 210 when it is three (3) or less slots (where each slot is of size sufficient to hold at least one column of SPE data) away from being completely empty; also shown is that in one embodiment "almost full" is set as five (5) slots rather than three (3), for reasons of practicality. Also shown is that in one embodiment pointer generator 218 treats as "almost empty" FIFO buffer 210 when it is twenty-seven (27) or less slots away from being completely full; also shown is that in one embodiment "almost empty" is set as twenty-eight (28) slots rather than twenty-seven (27), for reasons of practicality.

For sake of simplicity, the foregoing has demonstrated various almost empty and almost full settings for STS-1 frames. It will be appreciated by those in the art that the preceding examples need to be modified slightly for STS-N formats, since as described above the columns of STS-N frames will be N bytes (e.g., columns for STS-2 are 2 bytes wide, for STS-10 are 10 bytes wide, for STS 48 are 48 bytes wide, etc.).

The foregoing has described how to determine almost empty and almost full status of FIFO buffers when such buffers are keyed to non-standard SONET frames. Various embodiments use this information to determine when and whether positive and negative stuffing is to be done.

In various embodiments, with respect to FIFO buffer 210 (e.g., a buffer on the "receive" side of switch logic 202), pointer interpreter 208 examines the state of FIFO buffer 210 subsequent to writing sixteen columns (e.g., subsequent to writing 16 bytes for STS-1, 32 bytes for STS-2, 48 bytes for STS-3, etc.) of data to FIFO 210. The results of such examinations can be used in various embodiments ranging from "coarse" to "fine." In a coarse embodiment, the results of buffer examinations made just prior to a transition boundary between successive standard SONET frames received via ingress data link 112 are saved (because the large 27 column construct of the non-standard SONET is coincident with a period of time during which data may be being written to FIFO buffer 210 but not read out). If such buffer examinations associated with any two successive frames show that the buffer is in the "almost full" range at such transition points (e.g., two frames in succession end off with FIFO buffer 210 in an almost full condition), a decision is made for pointer interpreter 208 to engage in negative stuffing (and the pointers adjusted appropriately) as soon as practicable; in one embodiment, the most often negative stuffing can be done is once every four received standard SONET frame. In the more "fine" embodiments, a similar scheme to the foregoing is followed except that the decision to stuff is made on the basis of more than one examination before transitions between frames (e.g., based on the two examinations immediately preceding each transition point, which for two successive frames would result in the decision to stuff being based on 4 examinations of the buffer).

In various embodiments, with respect to FIFO buffer 210 (e.g., a buffer on the "receive" side of switch logic 202), pointer generator 212 examines the state of FIFO buffer 210 subsequent to reading sixteen columns (e.g., subsequent to reading 16 bytes for STS-1) of data from FIFO 210.

The results of such examinations can be used in various embodiments ranging from "coarse" to "fine." In a coarse embodiment, the results of buffer examinations made just prior to a transition boundary between successive standard SONET frame rows are saved because each row of a frame begins with a 3 column overhead group during which data may possibly not be written to FIFO buffer 210 by pointer interpreter 208 but during which pointer generator 212 may be trying to read data out. If such buffer examinations associated with any two successive frames show that the buffer is in the "almost empty" range at such transition points (i.e., the examinations show two successive frames end off with the buffer in the almost empty condition)), a decision is made for the pointer generator to engage in positive stuffing (and the pointers adjusted appropriately) as soon as practicable; in one embodiment, the most frequently positive stuffing can be done is once per every four standard SONET frames received into modified SONET node 200 via ingress data link 112. In the more "fine" embodiments, a similar scheme to the foregoing is followed except that the decision to stuff is made on the basis of more than one examination before transitions between rows (e.g., based on the three examinations immediately preceding each transition point, which for two successive rows would result in the decision to stuff being based on six examinations of the buffer).

In various embodiments, with respect to FIFO buffer 216 (e.g., a buffer on the "transmit" side of switch logic 202), pointer generator 214 examines the state of FIFO buffer 216 subsequent to writing sixteen columns (e.g., subsequent to writing 16 bytes for STS-1, 32 bytes for STS-2, 48 bytes for STS-3, etc.) of data to FIFO 216. The results of such examinations can be used in various embodiments ranging from "coarse" to "fine." In a coarse embodiment, the results of buffer examinations made just prior to a transition boundary between successive non-standard SONET frames received via from switch logic 202 are saved (because the large 27 column construct of the non-standard SONET denotes a time in which data will not typically be being written to FIFO buffer 216, but during which data may be taken out). If such buffer examinations associated with any two successive frames show that the buffer is in the "almost full" range at such transition points (e.g., two successive frames end off with an almost full buffer, a decision is made for pointer generator 214 to engage in negative stuffing (and the pointers adjusted appropriately) as soon as practicable; in one embodiment, the most often negative stuffing can be done is once for every four standard SONET frames transmitted out of modified node 200 over egress data link 114. In the more "fine"

embodiments, a similar scheme to the foregoing is followed except that the decision to stuff is made on the basis of more than one examination before transitions between frames (e.g., based on the two examinations immediately preceding each transition point, which for two successive frames would result in the decision to stuff being based on 4 examinations of the buffer).

In various embodiments, with respect to FIFO buffer 216 (e.g., a buffer on the "transmit" side of switch logic 202), pointer interpreter examines the state of FIFO buffer 216 subsequent to reading sixteen columns (e.g., subsequent to reading 16 bytes for STS-1) of data from FIFO 216. The results of such examinations can be used in various embodiments ranging from "coarse" to "fine." In a coarse embodiment, the results of buffer examinations made just prior to a transition boundary between successive standard SONET frames are saved because each row of a non-standard SONET frame begins with a 27 column overhead group during which data may possibly not be written to FIFO buffer 216 by pointer interpreter 214 but during which pointer generator 218 may be reading data. If such buffer examinations associated with any two successive frames show that the buffer is in the "almost empty" range at such transition points (i.e., pointer generator 218 ends off two successive frames with FIFO buffer 216 in an "almost empty" condition), a decision is made for pointer generator 218 to engage in positive stuffing (and the pointers adjusted appropriately) as soon as practicable; in one embodiment, the most frequently positive stuffing can be done is once per every four standard SONET frames transmitted from modified SONET node 200 via egress data link 114. In the more "fine" embodiments, a similar scheme to the foregoing is followed except that the decision to stuff is made on the basis of more than one examination before transitions between frames (e.g., based on the three examinations immediately preceding each transition point, which for two successive rows would result in the decision to stuff being based on six examinations of the buffer).

With reference now to Figure 5, depicted is a pictorial representation of a conventional data processing system which can be utilized in accordance with illustrative embodiments of the processes described herein. It should be noted that a graphical user interface systems (e.g., Microsoft Windows 98 or Microsoft Windows NT operating systems) and methods can be utilized with the data processing system depicted in Figure 5. Data processing system 520 is depicted which

includes system unit housing 522, video display device 524, keyboard 526, mouse 528, and microphone 548. Data processing system 520 may be implemented utilizing any suitable data processing system such as those manufactured and/or distributed by CISCO systems located in San Jose, California and Dallas, Texas (CISCO is a trademark of Cisco Systems, Incorporated; the data processing systems distributed and/or manufactured by Cisco also typically have associated with them switches and switching logic, such switches and switching logic including but not limited to optical switches for SONET), or Dell Computer Corporation located in Round Rock, Texas (DELL is a trademark of Dell Computer Corporation). Those skilled in the art will recognize that such conventional data processing systems will generally have a main data processing system board typically including by not limited to at least one processor (e.g., main processor, graphics processor, sound processor, etc.), at least one memory (RAM, DRAM, ROM) and at least one communications device (e.g., a network card or modem (optical or electronic)). In one embodiment, a Cisco Systems ONS (Optical Network System) 15900 Series Wavelength Router can be utilized in accordance with illustrative embodiments of the processes described herein. (CISCO, and "ONS 15900 Series Wavelength Router" are trademarks of Cisco Systems, Incorporated).

The foregoing detailed description has set forth various embodiments of the present invention via the use of block diagrams, flowcharts, and examples. It will be understood as notorious by those within the art that each block diagram component, flowchart step, and operations and/or components illustrated by the use of examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof. In one embodiment, the present invention may be implemented via logic encoded in a BIOS associated with a processor, while in another embodiment the present invention may be implemented via logic encoded in Application Specific Integrated Circuits (ASICs). However, those skilled in the art will recognize that the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard Integrated Circuits, as a computer program running on a computer, as firmware, or as virtually any combination thereof and that designing the circuitry and/or writing the code for the software or firmware would be well within the skill of one of ordinary skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the present invention are capable of

being distributed as a program product in a variety of forms, and that an illustrative embodiment of the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of a signal bearing media include but are not limited to the following: recordable type media such as floppy disks, hard disk drives, CD ROMs, digital tape, and transmission type media such as digital and analogue communication links using TDM or IP based communication links (e.g., packet links).

The foregoing described embodiments depict different components contained within, or connected with, different other components. It is to be understood that such depicted architectures are merely exemplary, and that in fact many other architectures can be implemented which achieve the same functionality. In an abstract, but still definite sense, any arrangement of components to achieve the same functionality is effectively "associated" such that the desired functionality is achieved. Hence, any two components herein combined to achieve a particular functionality can be seen as "associated with" each other such that the desired functionality is achieved, irrespective of architectures or intermedial components. Likewise, any two components so associated can also be viewed as being "operably connected", or "operably coupled", to each other to achieve the desired functionality.

Other embodiments are within the following claims.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those within the art that if a specific number of an introduced claim element is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles

"a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an"; the same holds true for the use of definite articles used to introduce claim elements. In addition, even if a specific number of an introduced claim element is explicitly recited, those skilled in the art will recognize that such recitation should typically be interpreted to mean at least the recited number (e.g., the bare recitation of "two elements," without other modifiers, typically means at least two elements, or two or more elements).

WHAT IS CLAIMED IS:

1 1. A method for determining buffer status, said method
2 comprising:
3 keying a buffer status to a transport gap other than a standard
4 SONET transport gap.

1 2. The method of Claim 1, wherein said keying a buffer
2 status to a transport gap other than a standard SONET transport gap
3 further comprises:
4 keying a transmit buffer status to a transport gap other than
5 the standard SONET transport gap.

1 3. The method of Claim 2, wherein said keying a transmit
2 buffer status to a transport gap other than the standard SONET
3 transport gap further comprises:
4 the transmit buffer interposed between a pointer interpreter
5 which receives data from a switching matrix and a pointer
6 generator which prepares a standard SONET STS-N frame.

1 4. The method of Claim 2, wherein said keying a transmit
2 buffer status to a transport gap other than the standard SONET
3 transport gap further comprises:
4 keying the transmit buffer to at least a column length of a
5 non-standard SONET transport gap.

1 5. The method of Claim 4, wherein each column of the non-
2 standard SONET transport gap contains 1 byte per each STS channel in
3 use.

1 6. The method of Claim 4, wherein said keying the transmit
2 buffer to at least a column length of a non-standard SONET transport
3 gap further comprises:
4 keying a pointer generator constructed to read data from the
5 transmit buffer to at least a column length of a non-
6 standard SONET transport gap.

1 7. The method of Claim 6, wherein said keying a pointer
2 generator constructed to read data from the transmit buffer to at

3 least a column length of a non-standard SONET transport gap further
4 comprises:

5 accepting input specifying a substantially 28-column almost-
6 empty range for the pointer generator reading data
7 constructed to read data from the transmit buffer.

1 8. The method of Claim 4, wherein said keying the transmit
2 buffer to at least a column length of a non-standard SONET transport
3 gap further comprises:

4 keying a pointer interpreter constructed to write data to the
5 transmit buffer to at least a column length of a non-
6 standard SONET transport gap.

1 9. The method of Claim 8, wherein said keying a pointer
2 generator constructed to write data to the transmit buffer to at
3 least a column length of a non-standard SONET transport gap further
4 comprises:

5 accepting input specifying a substantially 5-column almost-full
6 range for the pointer interpreter constructed to write
7 data to the transmit buffer.

1 10. The method of Claim 1, wherein said keying a buffer
2 status to a transport gap other than a standard SONET transport gap
3 further comprises:

4 keying a receive buffer status to a transport gap other than
5 the standard SONET transport gap.

1 11. The method of Claim 10, wherein said keying a receive
2 buffer status to a transport gap other than the standard SONET
3 transport gap further comprises:

4 the receive buffer interposed between a pointer generator which
5 feeds data to a switching matrix and a pointer
6 interpreter which receives data.

1 12. The method of Claim 10, wherein said keying a receive
2 buffer status to a transport gap other than the standard SONET
3 transport gap further comprises:
4 keying the receive buffer to at least a column length of a non-
5 standard SONET transport gap.

1 13. The method of Claim 12, wherein each column of the non-
2 standard SONET transport gap contains 1 byte per each STS channel in
3 use.

1 14. The method of Claim 12, wherein said keying the receive
2 buffer to at least a column length of a non-standard SONET transport
3 gap further comprises:
4 keying a pointer generator constructed to read data from the
5 receive buffer to at least a column length of a non-
6 standard SONET transport gap.

1 15. The method of Claim 14, wherein said keying a pointer
2 generator constructed to read data from the receive buffer to at
3 least a column length of a non-standard SONET transport gap further
4 comprises:
5 accepting input specifying a substantially 5-column almost-
6 empty range for the pointer generator constructed to read
7 data from the transmit buffer.

1 16. The method of Claim 12, wherein said keying the receive
2 buffer to at least a column length of a non-standard SONET transport
3 gap further comprises:
4 keying a pointer interpreter constructed to write data to the
5 receive buffer to at least a column length of a non-
6 standard SONET transport gap.

1 17. The method of Claim 16, wherein said keying the receive
2 buffer to at least a column length of a non-standard SONET transport
3 gap further comprises:
4 accepting input specifying a substantially 28-column almost-
5 full range for the pointer interpreter constructed to
6 write data to the receive buffer.

1 18. A system for determining buffer status, said system
2 comprising:
3 means for keying a buffer status to a transport gap other than
4 a standard SONET transport gap.

1 19. The system of Claim 18, wherein said means for keying a
2 buffer status to a transport gap other than a standard SONET
3 transport gap further comprises:
4 means for keying a transmit buffer status to a transport gap
5 other than the standard SONET transport gap.

1 20. The system of Claim 19, wherein said means for keying a
2 transmit buffer status to a transport gap other than the standard
3 SONET transport gap further comprises:
4 the transmit buffer interposed between a pointer interpreter
5 which receives data from a switching matrix and a pointer
6 generator which prepares a standard SONET STS-N frame.

1 21. The system of Claim 19, wherein said means for keying a
2 transmit buffer status to a transport gap other than the standard
3 SONET transport gap further comprises:
4 means for keying the transmit buffer to at least a column
5 length of a non-standard SONET transport gap.

1 22. The system of Claim 21, wherein each column of the non-
2 standard SONET transport gap contains 1 byte per each STS channel in
3 use.

1 23. The system of Claim 21, wherein said means for keying the
2 transmit buffer to at least a column length of a non-standard SONET
3 transport gap further comprises:
4 means for keying a pointer generator constructed to read data
5 from the transmit buffer to at least a column length of a
6 non-standard SONET transport gap.

1 24. The system of Claim 21, wherein said means for keying the
2 transmit buffer to at least a column length of a non-standard SONET
3 transport gap further comprises:

4 means for keying a pointer interpreter constructed to write
5 data to the transmit buffer to at least a column length
6 of a non-standard SONET transport gap.

1 25. The system of Claim 18, wherein said means for keying a
2 buffer status to a transport gap other than a standard SONET
3 transport gap further comprises:

4 means for keying a receive buffer status to a transport gap
5 other than the standard SONET transport gap.

1 26. The system of Claim 25, wherein said means for keying a
2 receive buffer status to a transport gap other than the standard
3 SONET transport gap further comprises:

4 the receive buffer interposed between a pointer generator which
5 feeds data to a switching matrix and a pointer
6 interpreter which receives data.

1 27. The system of Claim 25, wherein said means for keying a
2 receive buffer status to a transport gap other than the standard
3 SONET transport gap further comprises:

4 means for keying the receive buffer to at least a column length
5 of a non-standard SONET transport gap.

1 28. The system of Claim 27, wherein each column of the non-
2 standard SONET transport gap contains 1 byte per each STS channel in
3 use.

1 29. The system of Claim 27, wherein said means for keying the
2 receive buffer to at least a column length of a non-standard SONET
3 transport gap further comprises:

4 means for keying a pointer generator constructed to read data
5 from the receive buffer to at least a column length of a
6 non-standard SONET transport gap.

1 30. The system of Claim 27, wherein said means for keying the
2 receive buffer to at least a column length of a non-standard SONET
3 transport gap further comprises:
4 means for keying a pointer interpreter constructed to write
5 data to the receive buffer to at least a column length of
6 a non-standard SONET transport gap.

1 31. A SONET node comprising:
2 at least one pointer interpreter having an almost full buffer
3 detector set substantially equal to a number of columns
4 present in a non-standard SONET transport gap.

1 32. The SONET node of Claim 31, wherein the number of columns
2 present in a non-standard SONET transport gap comprises 27 columns of
3 data.

1 33. The SONET node of Claim 32, wherein each of the columns
2 comprises at least one byte of data for each STS channel in use.

1 34. The SONET node of Claim 31, wherein the SONET node
2 further comprises one or more components selected from the group
3 comprising a processor, a memory device, a bus, and a communications
4 device.

1 35. A SONET node comprising:
2 at least one pointer generator having an almost empty buffer
3 detector set substantially equal to a number of columns
4 present in a non-standard SONET transport gap.

1 36. The SONET node of Claim 35, wherein the number of columns
2 present in a non-standard SONET transport gap comprises 27 columns of
3 data.

1 37. The SONET node of Claim 36, wherein each of the columns
2 comprises at least one byte of data for each STS channel in use.

1 38. The SONET node of Claim 35, wherein the SONET node
2 further comprises one or more components selected from the group
3 comprising a processor, a memory device, a bus, and a communications
4 device.

1 39. A method for maintaining communications when using
2 asymmetrical gapping structures, said method comprising:
3 detecting a transition involving at least one SONET frame;
4 in response to said detecting yielding a determination that a
5 receive FIFO buffer is almost full during the transition
6 involving at least one SONET frame, engaging in negative
7 stuffing; and
8 in response to said detecting yielding a determination that a
9 receive FIFO buffer is almost empty during the transition
10 involving at least one SONET frame, engaging in positive
11 stuffing.

1 40. The method of Claim 39, wherein the determination that a
2 receive buffer is almost full comprises detecting that the receive
3 buffer has less empty space than that required to buffer data during
4 construction of a non-standard transport gap.

1 41. The method of Claim 40, wherein said detecting that the
2 receive buffer has less empty space than that required to buffer data
3 during construction of a non-standard transport gap further comprises
4 detecting that the receive buffer has space less than or equal to
5 twenty-eight columns of data when the non-standard transport gap is
6 twenty-seven columns of data in size.

1 42. The method of Claim 39, wherein the determination that a
2 receive buffer is almost empty comprises detecting that the receive
3 buffer has less empty space than that required to buffer data during
4 interpretation of a standard transport gap.

1 43. The method of Claim 42, wherein said detecting that the
2 receive buffer has less empty space than that required to buffer data
3 during interpretation of a standard transport gap further comprises
4 detecting that the receive buffer has space less than or equal to
5 five columns of data when the standard transport gap is three columns
6 of data in size.

1 44. A system for maintaining communications when using
2 asymmetrical gapping structures, said method comprising:
3 means for detecting a transition involving at least one SONET
4 frame;
5 means, responsive to said means for detecting yielding a
6 determination that a receive FIFO buffer is almost full
7 during the transition involving at least one SONET frame,
8 for engaging in negative stuffing; and
9 means, responsive to said means for detecting yielding a
10 determination that a receive FIFO buffer is almost empty
11 during the transition involving at least one SONET frame,
12 for engaging in positive stuffing.

1 45. The system of Claim 44, wherein said means, responsive to
2 said means for detecting yielding a determination that a receive FIFO
3 buffer is almost full during the transition involving at least one
4 SONET frame, for engaging in negative stuffing further comprises:
5 means for detecting that the receive buffer has less empty
6 space than that required to buffer data during
7 construction of a non-standard transport gap.

1 46. The system of Claim 45, wherein said means for detecting
2 that the receive buffer has less empty space than that required to
3 buffer data during construction of a non-standard transport gap
4 further comprises:
5 means for detecting that the receive buffer has space less than
6 or equal to twenty-eight columns of data when the non-
7 standard transport gap is twenty-seven columns of data in
8 size.

1 47. The system of Claim 44, wherein said means, responsive to
2 said means for detecting yielding a determination that a receive FIFO
3 buffer is almost empty during the transition involving at least one
4 SONET frame, for engaging in positive stuffing further comprises:
5 means for detecting that the receive buffer has less empty
6 space than that required to buffer data during
7 interpretation of a standard transport gap.

1 48. The system of Claim 47, wherein said means for detecting
2 that the receive buffer has less empty space than that required to
3 buffer data during interpretation of a standard transport gap further
4 comprises:
5 means for detecting that the receive buffer has space less than
6 or equal to five columns of data when the standard
7 transport gap is three columns of data in size.

1 49. A method for maintaining communications when using
2 asymmetrical gapping structures, said method comprising:
3 detecting a transition involving at least one SONET frame;
4 in response to said detecting yielding a determination that a
5 transmit FIFO buffer is almost full during the transition
6 involving at least one SONET frame, engaging in negative
7 stuffing; and
8 in response to said detecting yielding a determination that a
9 transmit FIFO buffer is almost empty during the
10 transition involving at least one SONET frame, engaging
11 in positive stuffing.

1 50. The method of Claim 49, wherein the determination that
2 the transmit buffer is almost full comprises detecting that the
3 transmit buffer has less empty space than that required to buffer
4 data during construction of a standard transport gap.

1 51. The method of Claim 50, wherein said detecting that the
2 transmit buffer has less empty space than that required to buffer
3 data during construction of a standard transport gap further
4 comprises detecting that the transmit buffer has space less than or
5 equal to five columns of data when the standard transport gap is
6 three columns of data in size.

1 52. The method of Claim 49, wherein the determination that
2 the transmit buffer is almost empty comprises detecting that the
3 transmit buffer has less empty space than that required to buffer
4 data during interpretation of a non-standard transport gap.

1 53. The method of Claim 52, wherein said detecting that the
2 transmit buffer has less empty space than that required to buffer
3 data during interpretation of a non-standard transport gap further
4 comprises detecting that the transmit buffer has space less than or
5 equal to twenty-eight columns of data when the non-standard transport
6 gap is twenty-seven columns of data in size.

1 54. A system for maintaining communications when using
2 asymmetrical gapping structures, said system comprising:
3 means for detecting a transition involving at least one SONET
4 frame;
5 means, responsive to said means for detecting yielding a
6 determination that a transmit FIFO buffer is almost full
7 during the transition involving at least one SONET frame,
8 for engaging in negative stuffing; and
9 means, responsive to said means for detecting yielding a
10 determination that a transmit FIFO buffer is almost empty
11 during the transition involving at least one SONET frame,
12 for engaging in positive stuffing.

1 55. The system of Claim 54, wherein said means, responsive to
2 said means for detecting yielding a determination that a transmit
3 FIFO buffer is almost full during the transition involving at least
4 one SONET frame, for engaging in negative stuffing further comprises:
5 means for detecting that the transmit buffer has less empty
6 space than that required to buffer data during
7 construction of a standard transport gap.

1 56. The system of Claim 55, wherein said means for detecting
2 that the transmit buffer has less empty space than that required to
3 buffer data during construction of a standard transport gap further
4 comprises:
5 means for detecting that the transmit buffer has space less
6 than or equal to five columns of data when the standard
7 transport gap is three columns of data in size.

1 57. The system of Claim 54, wherein said means, responsive to
2 said means for detecting yielding a determination that a transmit
3 FIFO buffer is almost empty during the transition involving at least
4 one SONET frame, for engaging in positive stuffing further comprises:
5 means for detecting that the transmit buffer has less empty
6 space than that required to buffer data during
7 interpretation of a non-standard transport gap.

1 58. The system of Claim 57, wherein said means for detecting
2 that the transmit buffer has less empty space than that required to
3 buffer data during interpretation of a non-standard transport gap
4 further comprises:
5 means for detecting that the transmit buffer has space less
6 than or equal to twenty-eight columns of data when the
7 non-standard transport gap is twenty-seven columns of
8 data in size.

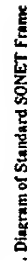
BUFFER STATUS IN AN ASYMMETRICAL GAP ENVIRONMENT

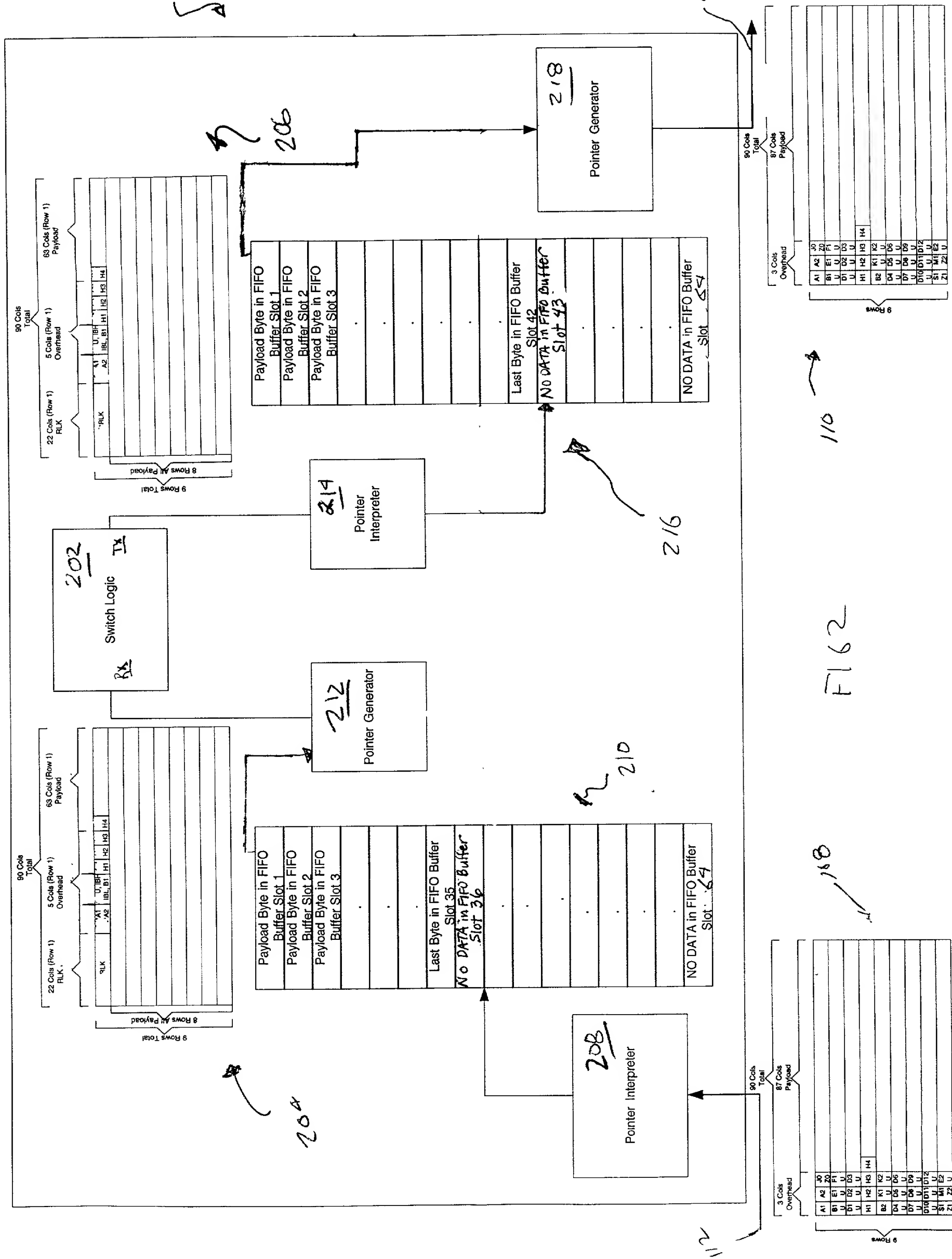
Robert A. Hall
Stephen P. Kolecki

5 ABSTRACT

10 A method for determining buffer status, the method including
but not limited to keying a buffer status to a transport gap other
than a standard SONET transport gap. In various embodiments,
hardware and/or software are utilized to effect the foregoing-
described method. A related method for maintaining communications
when using asymmetrical gapping structures, the method including but
not limited to detecting a transition involving at least one SONET
frame; engaging in negative stuffing, in response to the detecting
yielding a determination that a receive FIFO buffer is almost full
15 during the transition involving at least one SONET frame; and
engaging in positive stuffing, in response to the detecting yielding
a determination that a receive FIFO buffer is almost empty during the
transition involving at least one SONET frame. In various
embodiments, hardware and/or software are utilized to effect the
20 foregoing-described method. A related method for maintaining
communications when using asymmetrical gapping structures, the method
including but not limited to detecting a transition involving at
least one SONET frame; engaging in negative stuffing, in response to
said detecting yielding a determination that a transmit FIFO buffer
25 is almost full during the transition involving at least one SONET
frame; and engaging in positive stuffing, in response to said
detecting yielding a determination that a transmit FIFO buffer is
almost empty during the transition involving at least one SONET
frame. In various embodiments, hardware and/or software are utilized
30 to effect the foregoing-described method.

631883 v4





"Receive" Side

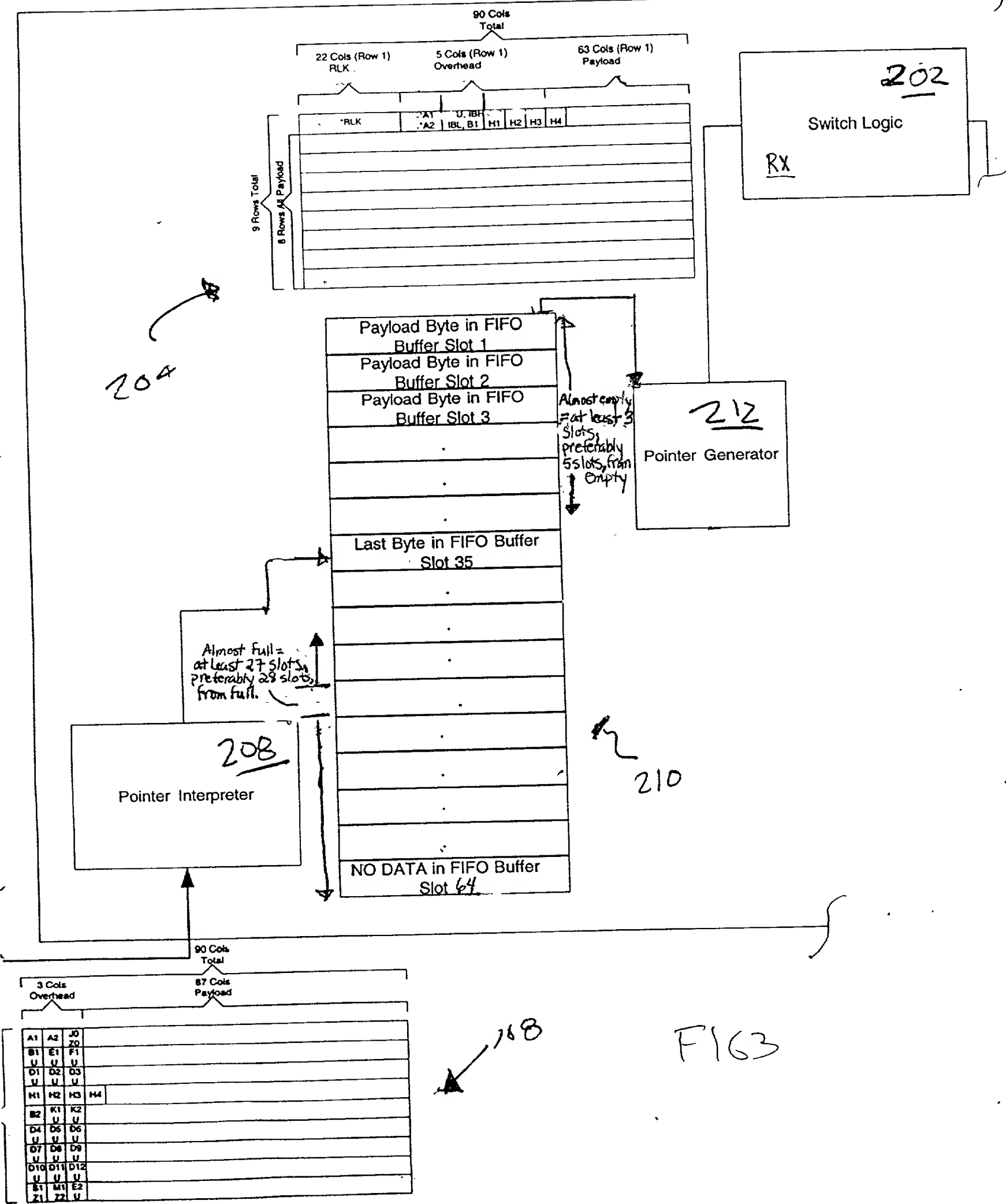
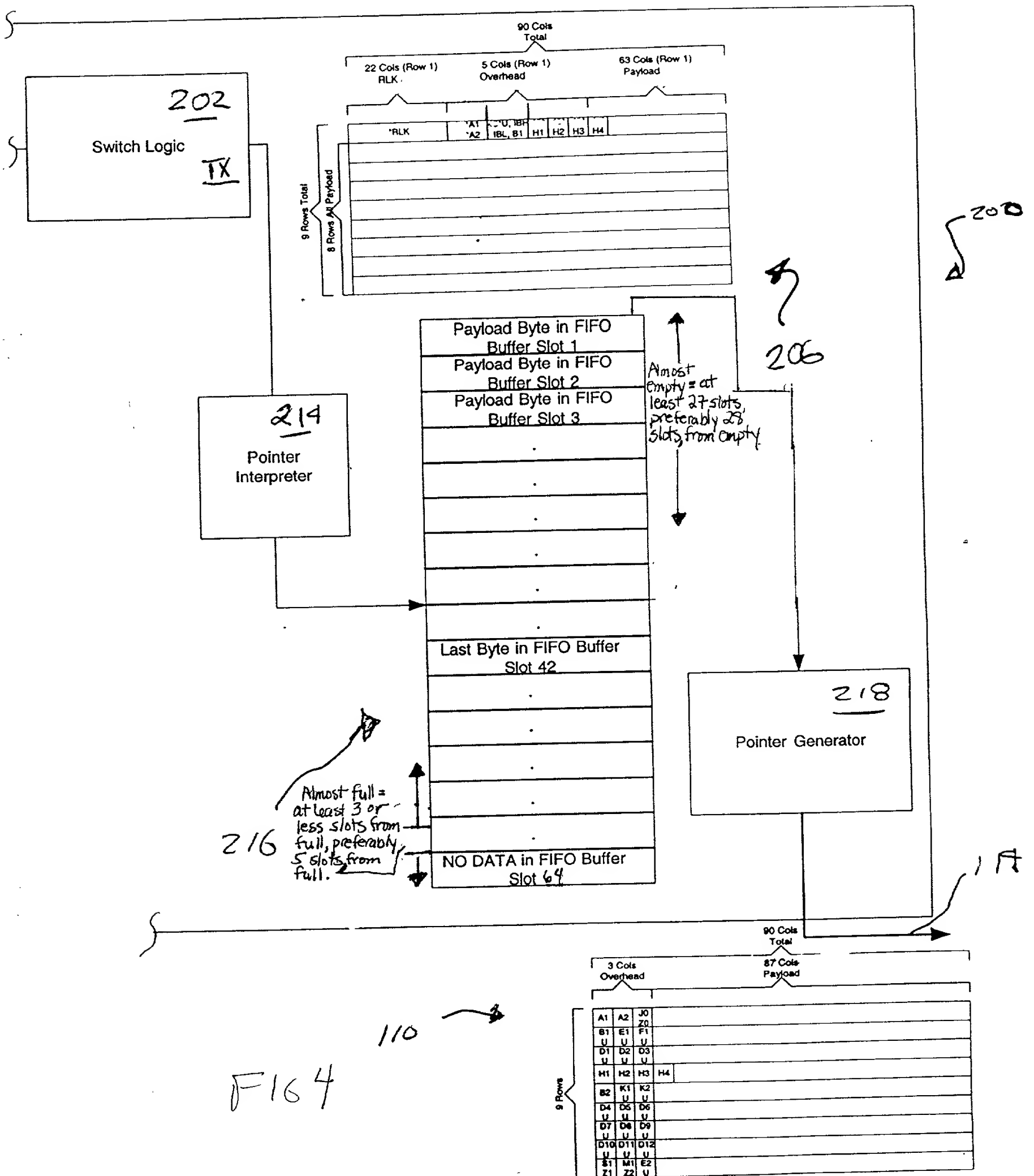


FIG 3

"TRANSMIT" Side



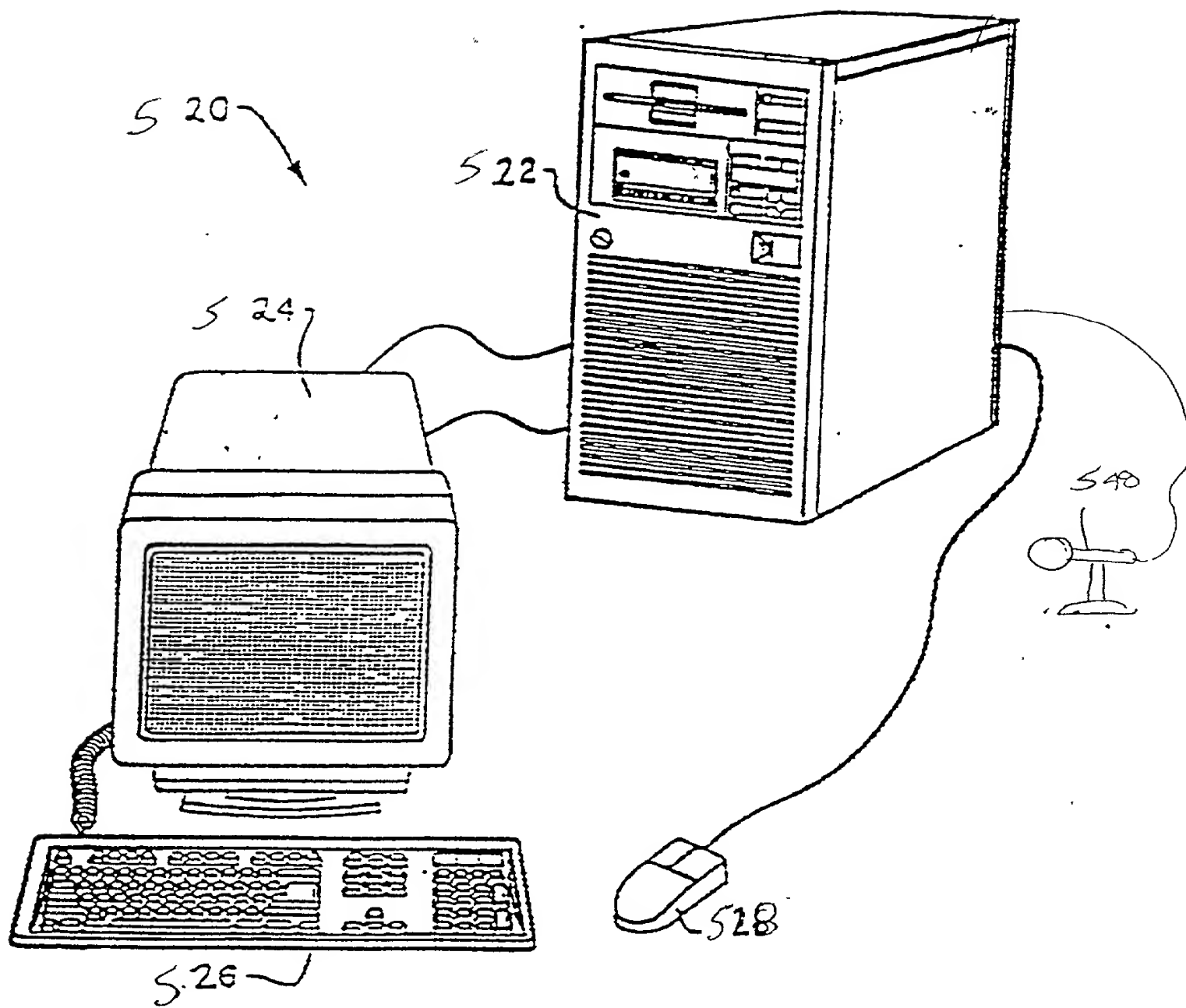


FIG 5

DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below adjacent to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of subject matter (process, machine, manufacture, or composition of matter, or an improvement thereof) which is claimed and for which a patent is sought by way of the application entitled

Buffer Status In An Asymmetrical Gap Environment

which (check) ☒ is attached hereto.
☐ and is amended by the Preliminary Amendment attached hereto.
☐ was filed on _____ as Application Serial No. _____
☐ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information, which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) of any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
Number	Country	Day/Month/Year Filed	Yes	No
N/A			<input type="checkbox"/>	<input type="checkbox"/>

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below:

Provisional Application Number	Filing Date
N/A	

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) or PCT international application(s) designating the United States of America listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior application(s) in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information, which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56, which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

Application Serial No.	Filing Date	Status (patented, pending, abandoned)
N/A		

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

Alan H. MacPherson (24,423); Brian D. Ogonowsky (31,988); David W. Heid (25,875); Norman R. Klivans (33,003); Edward C. Kwok (33,938); David E. Steuber (25,557); Michael Shenker (34,250); Stephen A. Terrile (32,946); Peter H. Kang (40,350); Ronald J. Meertin (29,089); Ken John Koestner (33,004); Omkar K. Suryadevara (36,320); David T. Millers (37,396); Michael P. Adams (34,763); Robert B. Morrill (43,817); Michael J. Halbert (40,633); Gary J. Edwards (41,008); James E. Parsons (34,691); Daniel P. Stewart (41,332); Phillip W. Woo (39,880); John T. Winburn (26,822); Tom Chen (42,406); Fabio E. Marino (43,339); Don C. Lawrence (31,975); Marc R. Ascolese (42,268); Carmen C. Cook (42,433); David G. Dolezal (41,711); Roberta P. Saxon (43,087); Mary Jo Bertani (42,321); Dale R. Cook (42,434); Sam G. Campbell (42,381); Matthew J. Brigham (44,047); Hugh H. Matsubayashi (43,779); Patrick D. Benedicto (40,909); T.J. Singh (39,535); Shireen Irani Bacon (40,494); Rory G. Bens (44,028); George Wolken, Jr. (30,441); John A. Odozynski (28,769); Cameron K. Kerrigan (44,826); Paul E. Lewkowicz (44,870); Theodore P. Lopez (44,881); Mayankkumar M. Dixit (44,064); Eric Stephenson (38,321); Christopher Allenby (45,906); David C. Hsia (46,235); and Mark J. Rozman (42,117).

Please address all correspondence and telephone calls to:

Dale R. Cook
Attorney for Applicants
SKJERVEN, MORRILL, MacPHERSON, FRANKLIN & FRIEL LLP
25 Metro Drive, Suite 700
San Jose, California 95110-1349

Telephone: 512-794-3600
Facsimile: 512-794-3601

I declare that all statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and all statements made herein are made with the knowledge that whoever, in any matter within the jurisdiction of the Patent and Trademark Office, knowingly and willfully falsifies, conceals, or covers up by any trick, scheme, or device a material fact, or makes any false, fictitious or fraudulent statements or representations, or makes or uses any false writing or document knowing the same to contain any false, fictitious or fraudulent statement or entry, shall be subject to the penalties including fine or imprisonment or both as set forth under 18 U.S.C. 1001, and that violations of this paragraph may jeopardize the validity of the application or this document, or the validity or enforceability of any patent, trademark registration, or certificate resulting therefrom.

Full name of sole (or first joint) inventor: Robert A. Hall

Inventor's Signature: Robert A. Hall

Date: 5-31-00

Residence: Richardson, Texas

Post Office Address: 4405 Southpointe Dr.

Citizenship: United States

Richardson, Texas 75082

Full name of second inventor: Stephen P. Kolecki

Inventor's Signature: Stephen P. Kolecki

Date: 5/31/00

Residence: Richardson, Texas

Post Office Address: 280 W. Renner Rd.

Citizenship: United States

Richardson, Texas 75080